

# Conception robotique avec ROS

Référence : PYRT020

Durée : 3 jours

Certification : **Aucune**

## CONNAISSANCES PREALABLES

- Connaissance générale des systèmes d'informations, d'un langage de développement comme Python ou C++.

## PROFIL DES STAGIAIRES

- Chefs de projet robotique, développeurs et toute personne souhaitant utiliser le système ROS pour concevoir une application robotique..

## OBJECTIFS

- Comprendre l'architecture de ROS. • Savoir choisir et mettre en oeuvre les composants adaptés à un projet robotique. • Les travaux pratiques sont réalisés avec ROS 2.

## CERTIFICATION PREPAREE

Aucune

## METHODES PEDAGOGIQUES

- Mise à disposition d'un poste de travail par stagiaire
- Remise d'une documentation pédagogique papier ou numérique pendant le stage
- La formation est constituée d'apports théoriques, d'exercices pratiques, de réflexions et de retours d'expérience
- Le suivi de cette formation donne lieu à la signature d'une feuille d'émargement

## FORMATEUR

Consultant-Formateur expert Robotique et IA

## METHODE D'EVALUATION DES ACQUIS

- Auto-évaluation des acquis par le stagiaire via un questionnaire
- Attestation de fin de stage adressée avec la facture

## CONTENU DU COURS

### Introduction

- Présentation de ROS (Robot Operating System), historique, contraintes, fonctionnalités, licence
- Objectifs des créateurs de ROS
- Point sur les dernières distributions de ROS
- Tour d'horizon des robots compatibles
- Exemples d'utilisation de ROS

### Architecture

- Principe de fonctionnement de ROS : modularité, légèreté, compatibilité
- Les différents éléments : les nœuds, le master, les topics, les messages, les services
- L'organisation des fichiers, l'installation : les packages, les stacks
- Présentation packages disponibles

### Mise en oeuvre

- Installation sur Ubuntu
- Configuration de l'environnement
- Présentation du filesystem ROS et de roscd, eosis, rospack
- Premier pas : création d'un package
- Gestion des nœuds : utilisation de roscore, rosnodet et rosrund
- Création de topics et de services

### Développement

- Programmation en python d'outils de publication et réception de messages (topics) et de services et clients simples

### Exploitation

- Utilisation de fichiers .bag pour stocker des messages et les réutiliser
- Gestion des dépendances entre packages

- Exécution de ROS sur une infrastructure de plusieurs serveurs
- Présentation des librairies principales

### **Extensions**

- Présentation des librairies principales